

Traccia:

Abbiamo già parlato del buffer overflow, una vulnerabilità che è conseguenza di una mancanza di controllo dei limiti dei buffer che accettano input utente.

Nelle prossime slide vedremo un esempio di codice in C volutamente vulnerabile ai BOF, e come scatenare una situazione di errore particolare chiamata «segmentation fault», ovvero un errore di memoria che si presenta quando un programma cerca inavvertitamente di scrivere su una posizione di memoria dove non gli è permesso scrivere (come può essere ad esempio una posizione di memoria dedicata a funzioni del sistema operativo).

BOF Buffer 10

Script

```
File Actions Edit View Help
GNU nano 8.0
#include <stdio.h>

int main () {
char buffer [10];

printf ("Si prega di inserire il nome utente")
scanf ("%s", buffer)

printf ("Nome utente inserito: %s\n", buffer)

return 0;
}
```

Esecuzione programma

```
(kali@kali)-[~/Desktop]
$ ./BOF
Si prega di inserire il nome utente:123456789012345678
Nome utente inserito: 123456789012345678
zsh: segmentation fault ./BOF
```

BOF Buffer 20

Script

```
File Actions Edit View Help
GNU nano 8.0
#include <stdio.h>

int main () {
char buffer [20];

printf ("Si prega di inserire il nome utente:");
scanf ("%s", buffer);

printf ("Nome utente inserito: %s\n", buffer);

return 0;
}
```

Esecuzione programma

```
(kali㉿kali)-[~/Desktop]
$ ./BOF
Si prega di inserire il nome utente:12345678901234567890102345678901234567890
Nome utente inserito: 12345678901234567890102345678901234567890
zsh: segmentation fault ./BOF
```

BOF Buffer 30

Script

```
GNU nano 8.0
#include <stdio.h>

int main () {
char buffer [30];

printf ("Si prega di inserire il nome utente:");
scanf ("%s", buffer);

printf ("Nome utente inserito: %s\n", buffer);

return 0;
}
```

Esecuzione Programma

```
(kali㉿kali)-[~/Desktop]  
$ ./BOF  
Si prega di inserire il nome utente:1234567890123456789012345678901234567890  
Nome utente inserito: 1234567890123456789012345678901234567890  
zsh: segmentation fault ./BOF
```